REMARKS

Applicant has received the Office action dated October 7, 2005, in which the Examiner: 1) rejected claims 1, 12, 19 and 29 as allegedly indefinite; 2) rejected claims 12-28 as allegedly anticipated by Trainin (U.S. Pub. No. 2002/0144073); and 3) rejected claims 1, 3-11, 29-30 and 32-39 as allegedly unpatentable over Trainin in view of Steele (U.S. Pub. No. 2001/0056420).

With this Response, Applicant amends claims 12, 19, 33 and 39, cancels claims 34-38, and presents new claims 40-42. Reconsideration is respectfully requested.

I. SECTION 112 REJECTIONS

The Office action dated October 7, 2005 alleges that claims 1, 12, 19 and 29 suffer under Section 112, second paragraph, which Applicant respectfully traverses. In particular, the Office action states:

> It is not clear how software can perform memory operations without involving any hardware and it is not clear how hardware can perform the return of memory blocks without involving the use of software.

(Office action dated October 7, 2005). There is no recitation in claims 1, 12, 19 and 29 that software should operate without hardware, and that hardware should operate without software. In fact, illustrative claim 29 specifically recites, "a microprocessor executing a software stream... ."

Applicant acknowledges that the Office uses the broadest reasonable interpretation in examining claims. However, Applicant submits that in view of the specification and knowledge of one of ordinary skill it is an unreasonable interpretation that there is no "distinction between performing memory operations by a software stream and returning memory blocks by a hardware device." (Office action dated October 7, 2005, numbered paragraph 1). The specification and claims use the term "software stream" or "software thread" in relation to software executed on the CPU. (Specification Paragraph [0009]; Figure 1;

claim 29). By contrast, with regard to "hardware device," the specification clearly states:

> A non-limiting list of hardware devices that could implement the heap memory management method comprises graphics cards, network interface cards, audio devices, and mass storage such as hard drives and compact disc drives.

(Specification Paragraph [0047]; *see also* claims 33-39). If hypothetically the interpretation is accepted that there is no "distinction between performing memory operations by a software stream and returning memory blocks by a hardware device," the interpretation renders redundant and non-sensical the specification and claims, and thus such an interpretation is clearly unreasonable.

Other issued patents, evidencing the knowledge of one of ordinary skill, recognize the difference between software threads and hardware.

> FIG. 3 illustrates an exemplary queue data structure 42 which can be accessed concurrently by cache-coherent entities **(e.g., software threads or processors)** and a non-cache-coherent entity **(e.g., hardware)**.

(U.S. Pat. No. 6,687,927, Col. 4, lines 65 – Col. 5, line 2 (emphasis added)).

Thus, Applicant respectfully submits that the interpretation that there is no distinction between "software streams" and "hardware devices" in the Office action dated October 7, 2005 is not a reasonable interpretation, and the rejection should be withdrawn.

II. RESPONSE TO ARGUMENTS

In the Response to Office action filed June 22, 2005, the Applicant made the statement, "As the Examiner is no doubt aware, hardware devices such as graphics cards, network interface cards, audio devices and mass storage devices are not part of the cache coherency domain of the main processor... ." In response, the Office action dated October 7, 2005 states, "The Examiner is not aware of teachings that support this assertion." Although discussed above with respect to distinctions between software threads and hardware, the issued patent also sheds light on the knowledge of one of ordinary skill with respect to cache-coherent and non-cache-coherent entities.

> FIG. 3 illustrates an exemplary queue data structure 42 which can be accessed concurrently by **cache-coherent entities** (e.g., software threads or processors) and a **non-cache-coherent entity** (e.g., hardware). **Cache-coherent entities are entities which participate in the same cache coherency protocol. Non-cache-coherent entities, on the other hand, do not participate in the same cache coherency scheme as the cache-coherent entities (i.e., the host processor(s)) in the processor-based device.**

(U.S. Pat. No. 6,687,927, Col. 4, lines 65 – Col. 5, line 7 (emphasis added)). With the illustrative linked list of the current specification software streams within the coherency domain perform atomic operations on a first end of the linked list, and a hardware device returns memory blocks to the second end of the linked list in spite of the fact the hardware device is not capable of performing, or is not allowed to perform, atomic operations in a cache coherency domain to which it does not belong.

III. ART-BASED REJECTIONS

A. Claim 1

The Office action dated April 28, 2005 rejects the claims as allegedly obvious over Trainin in view of Steele.

Trainin is directed to a method for memory heap management and buddy system management for service aware networks. (Trainin Title.) While Trainin may disclose free memory block management by way of a linked list (*See, e.g.*, Trainin Paragraph [0036]), the discussion of use of the linked list appears to be only for software on the Trainin CPU. (*See, e.g.*, Trainin Detailed Description; Abstract). Steele is directed to a lock-free implementation of a concurrent shared object with dynamic node allocation and a distinguishing pointer value. (Steele Title.) The Steele reference appears to be concerned only with concurrency between software streams executing within the same cache coherency domain, and thus does not even contemplate issues addressed by the current specification.

Claim 1, by contrast, specifically recites, "performing, **by a software stream**, heap memory operations on a first end of a linked list of free heap memory of a heap pile; and concurrently returning a return block of heap memory, **by a hardware device that used the return block of heap memory,** to the heap pile at a second end of the linked list of free heap memory." Applicant respectfully submits that Trainin and Steele fail to teach or fairly suggest that a hardware device could or should return a block of heap memory concurrently with operations of a software stream. For this reason alone claim 1 should be allowed.

Moreover, Trainin and Steele appear to be silent as to any difference in operation as between software streams and hardware devices, and thus fail to teach or suggest "performing, by a software stream, heap memory operations **on a first end of a linked list** ... [and] **returning** a return block of heap memory, by a hardware device that used the return block of heap memory, to the heap pile **at a second end of the linked list** ... ."

For these reasons, Applicant respectfully submits that claim 1 is not rendered unpatentable by Trainin and Steele, and that claim 1 should be allowed together with all claims that depend from claim 1 (claims 3-11 and 40-42).

B. Claim 12

Claim 12 stands rejected as allegedly anticipated by Trainin. Applicant amends claim 12 to more clearly define over the system of Trainin.

Claim 12 specifically recites, "maintaining unused blocks of heap memory as a linked list, and wherein the unused blocks of the linked list comprise a first block at a beginning of the linked list, a second block pointed to the first block, and a third block at an end of the linked list; **removing, by a software stream using an atomic operation, the first block from the linked list**, thus making the second block the beginning of the linked list; and **returning a return block, by a hardware device that used the return block, to the linked list by placing**

the return block at the end of the linked list with a non-atomic operation."[1] Applicant respectfully submits that Trainin fails to expressly or inherently teach that a hardware device could or should return a block of heap memory to a linked list that is also accessible by a software stream. Moreover, Applicant respectfully submits that Trainin fails to teach or fairly suggest any distinction between software streams using an atomic operation regarding the linked list, and hardware devices using non-atomic operations regarding the linked list.

For these reasons, Applicant respectfully submits that claim 12 is not rendered unpatentable by Trainin, and that claim 12 should be allowed together with all claims that depend from claim 12 (claims 13-18).

C. Claim 19

Claim 19 stands rejected as allegedly anticipated by Trainin. Applicant amends claim 19 to more clearly delineate the processor that executes the software thread from the claimed hardware device.

Claim 19 recites, "allowing **a software thread executed on a processor** to add and remove blocks of heap memory from a linked list of free blocks of heap memory in a last-in/first-out (LIFO) fashion at a first end of the linked list; and allowing a hardware device that uses blocks of heap memory to add the blocks of heap memory to the linked list of free blocks of heap memory at a second end of the linked list, **the hardware device coupled to the processor by way of a communication bus.**" Applicant respectfully submits that Trainin fails to expressly or inherently teach that a hardware device (different than the processor on which the software thread executes) could or should return a block of heap memory to a linked list that is also accessible by the software thread.

For these reasons, Applicant respectfully submits that claim 19 is not rendered unpatentable by Trainin, and that claim 19 should be allowed together with all claims that depend from claim 19 (claims 20-28).

[1] *See, e.g.*, Specification Paragraph [0049]. If only a single hardware agent is allowed to return blocks, the operation need not be atomic

D. Claim 29

Claim 29 stands rejected as allegedly obvious over Trainin and Steele.

Claim 29 specifically recites, "**a microprocessor executing a software stream;** ... a first bridge logic device coupling the microprocessor to the main memory array; **a hardware device coupled to the heap memory through the first bridge logic device**; wherein **the software stream executed on the microprocessor removes blocks of heap memory from a beginning of the heap pile; and simultaneously the hardware device returns blocks of heap memory used by the hardware device to an end of the heap pile**." The hardware device is separately claimed from the microprocessor that executes the software stream. Thus, the suggestion of the Office action that the processor and software executed on the processor of Trainin fulfill the claim limitations regarding the hardware device and software stream elements falls woefully short. Thus, even if hypothetically the teachings of Steele are precisely as the Office action suggests (which Applicant does not admit), Trainin and Steele still fail to teach or fairly suggest "the software stream executed on the microprocessor removes blocks of heap memory from a beginning of the heap pile; and simultaneously the hardware device returns blocks of heap memory used by the hardware device to an end of the heap pile."

For these reasons, Applicant respectfully submits that claim 29 is not rendered unpatentable by Trainin and Steele, and that claim 29 should be allowed together with all claims that depend from claim 29 (claims 30 and 32-33). Applicant cancels claims 34-39 to reduce the number of claims, and place their limitations in claim 33.

E. Claim 39

Claim 39 stands rejected as allegedly as allegedly obvious over Trainin and Steele. Applicant amends claim 39 to more clearly delineate the difference between the processor that executes the software stream and the claimed hardware device.

Claim 39 specifically recites, "performing, by a software stream executed on a processor, heap memory operations on a first end of a linked list of free heap memory of a heap pile; and concurrently returning a return block of heap memory to the heap pile at a second end of the linked list of free heap memory, the returning by a hardware device coupled to the processor selected from group consisting of a graphics card, a network interface card, an audio device or a mass storage device." The hardware device is separately called out from the processor that executes the software stream. Thus, the suggestion of the Office action that the processor and software executed on the processor of Trainin fulfill the claim limitations regarding the hardware device and software stream elements falls woefully short. Thus, even if hypothetically the teaches of Steele are precisely as the Office action suggests (which Applicant does not admit), Trainin and Steele still fail to teach or fairly suggest the limitations of claim 39.

For these reasons, Applicant respectfully submits that claim 39 is not rendered unpatentable by Trainin and Steele, and that claim 39 should be allowed.

IV. NEW CLAIMS

With this Response, Applicant presents new claims 40-42. Applicant respectfully submits these claims are allowable for at least the same reasons as claim 1 from which they depend, as well as the additional limitations therein.

V. CONCLUSION

In the course of the foregoing discussions, Applicant may have at times referred to claim limitations in shorthand fashion, or may have focused on a particular claim element. This discussion should not be interpreted to mean that the other limitations can be ignored or dismissed. The claims must be viewed as a whole, and each limitation of the claims must be considered when determining the patentability of the claims. Moreover, it should be understood that there may be other distinctions between the claims and the cited art which have yet to be raised, but which may be raised in the future.

Applicant respectfully requests reconsideration and that a timely Notice of Allowance be issued in this case. It is believed that no extensions of time or fees are required, beyond those that may otherwise be provided for in documents accompanying this paper. However, in the event that additional extensions of time are necessary to allow consideration of this paper, such extensions are hereby petitioned under 37 C.F.R. § 1.136(a), and any fees required (including fees for net addition of claims) are hereby authorized to be charged to Hewlett-Packard Development Company's Deposit Account No. 08-2025.

Respectfully submitted,

Mark E. Scott
PTO Reg. No. 43,100
CONLEY ROSE, P.C.
(713) 238-8000 (Phone)
(713) 238-8008 (Fax)
ATTORNEY FOR APPLICANT

HEWLETT-PACKARD COMPANY
Intellectual Property Administration
Legal Dept., M/S 35
P.O. Box 272400
Fort Collins, CO 80527-2400